Contingut

Realitza els exercicis de manera individual i comenta en el fòrum les teues solucions. Crea una BD per a cada BD lògica

Si els noms de les taules o els atributs tenen caràctes no vàlids, cal adaptar-los. D'altra banda, per si esteu treballant amb MariaDB o en MySql no us està creant per defecte les taules com Innodb. Per a assegurar-nos, acabarem la instruccions amb `engine=innodb;`. Amb això ens assegurem de crear BD que implementen la integritat referencial.

Sol haver-hi diverses maneres d'especificar-ho possibles. Una solució pot ser vàlida encara que no siga la donada ací.

1. EXERCICI 1

Tradueix el següent **Model Lògic Relacional** a **Model Físic en SQL**:

Pel·lícules (cod_pel, nom, director, any, genere, visionada) CP: {cod_pel} VNN: {nom}	**Dominios**: cod_pel: sencer > 0 nom: cadena(50) director: cadena(50) data: data genere: 'suspens', 'acció', 'terror', 'comèdia', 'drama' o 'cienciafic' visionada: 'si' o 'no'

```
CREATE SCHEMA EXER1;
USE EXER1;
CREATE TABLE PELICULES(
  cod_pel INTEGER CHECK (cod_pel > 0),
  nom VARCHAR(50) NOT NULL,
  director VARCHAR(50),
  any  date,
  genero varchar(10) CHECK (genero IN
('suspens','acció','terror','comèdia','drama','cienciafic')),
  visionada varchar(2) CHECK (visionada IN ('si','no')),
  CONSTRAINT PEL_COD_PK PRIMARY KEY(cod_pel)
) engine=innodb;
```

Altra opció en el tipus ENUM:

```
CREATE SCHEMA EXER1;
USE EXER1;

CREATE TABLE PELICULES(
  cod_pel INTEGER CHECK (cod_pel > 0),
  nom VARCHAR(50) NOT NULL,
  director VARCHAR(50),
  any  date,
  genero ENUM('suspens','acció','terror','comèdia','drama','cienciafic'),
  visionada ENUM('si','no'),
  CONSTRAINT PEL_COD_PK PRIMARY KEY(cod_pel)
) engine=innodb;
```

2. EXERCICI 2

Tradueix el següent **Model Lògic Relacional** a **Model Físic en SQL**:

País (nom_p, bandera, renda) CP: {nom_p} **Ciutat** (nom_c, habitants, nom_p) CP: {nom_c, nom_p} CAj: {nom_p} → País	**Dominis**: nom_p: cadena(30) bandera: cadena(70) renda: coma flotant nom_c: cadena(40) habitantes: enter

```
CREATE SCHEMA EXER2;
USE EXER2;

CREATE TABLE PAIS (
  nom_p VARCHAR(30),
  bandera VARCHAR(70),
  renda FLOAT,
  constraint pai_nom_pk PRIMARY KEY (nom_p)
)engine=innodb;

CREATE TABLE CIUDAD (
  nom_c VARCHAR(40),
  habitants INTEGER,
  nom_p VARCHAR(30),
  constraint ciu_nom_pk PRIMARY KEY  (nom_c,nom_p),
  constraint ciu_nom_fk FOREIGN KEY (nom_p) REFERENCES PAIS (nom_p)
)engine=innodb;
```

Altra opció:

```
CREATE SCHEMA EXER2;
USE EXER2;

CREATE TABLE PAIS (
  nom_p VARCHAR(30)primary key,
  bandera VARCHAR(70),
  renda FLOAT,
  )engine=innodb;
```

BUTLLETÍ.A.MODEL FÍSIC DDL

```
CREATE TABLE CIUDAD (
  nom_c VARCHAR(40),
  habitants INTEGER,
  nom_p VARCHAR(30) REFERENCES PAIS (nom_p),
  constraint ciu_nom_pk PRIMARY KEY  (nom_c,nom_p)
) engine=innodb;
```

3. EXERCICI 3

Tradueix el següent **Model Lògic Relacional** a **Model Físic en SQL**:

Plet (cod_plet, resultat) CP: {cod_plet} **Recurs** (cod_rec, data, cod_plet) CP: {cod_rec, cod_plet} CAj: {cod_plet} → Plet	**Dominis**: cod_plet: enter resultat: 'culpable', 'innocent' i 'sobresegut' cod_rec: entero data: data

```
CREATE SCHEMA EXER3;
USE EXER3;

CREATE TABLE PLET (
  cod_plet INTEGER,
  resultat VARCHAR(10) CHECK (resultat IN ('culpable','inocente','sobreseido')),
  constraint ple_cod_pk PRIMARY KEY (cod_plet)
) engine=innodb;
CREATE TABLE RECURS (
  cod_rec INTEGER,
  any DATE,
  cod_plet INTEGER,
  constraint rec_cod_pk PRIMARY KEY  (cod_rec,cod_plet),
  constraint rec_cod_fk FOREIGN KEY (cod_plet) REFERENCES plet (cod_plet))
engine=innodb;
```

4. EXERCICI 4

Tradueix el següent **Model** a **Model Físic en SQL** donant nom a totes les restriccions:

Empleat (cod_emp, telf) CP {cod_emp} Client (dni, nom, cognoms) CP: {dni} **Cotxe** (n_bastidor, marca, model, color, matricula) CP: {n_bastidor} Únic: {matrícula} **Venda** (cod_emp, dni, n_bastidor, data, preu) CP: {n_bastidor, dni}	**Dominis:** cod_emp: enter telf: cadena(9) dni: cadena(9) nom: cadena(30) cognoms: cadena(60) n_bastidor: cadena(17) marca: cadena(20) model: cadena(15) color: 'roig', 'blanc', 'blau', 'negre', 'plata',

VNN: {cod_emp} VNN: {fecha} VNN: {preu} CAj: {cod_emp} → Empleat CAj: {dni} → Client CAj: {n_bastidor} → Cotxe	'grisa', 'groc' i 'verd' matricula: cadena(7) cod_emp: enter data: data **preu: sencer > 0**

```
CREATE TABLE EMPLEAT (
  cod_emp INTEGER,
  telf VARCHAR(9),
  constraint emp_cod_pk PRIMARY KEY (cod_emp)
) engine=innodb;

CREATE TABLE CLIENTE (
  dni VARCHAR(9),
  nombre VARCHAR(30),
  apellidos VARCHAR(60),
  constraint cli_dni_pk PRIMARY KEY (dni)
) engine=innodb;

CREATE TABLE COCHE (
  n_bastidor VARCHAR(17),
  marca VARCHAR(20),
  modelo VARCHAR(15),
  color VARCHAR(8)
--- TAMBÉ SERIA VÀLID ENUM
  constraint coc_chk CHECK (color IN ('roig', 'blanc', 'blau', 'negre',
'plata',,'grisa','groc','verd')),
  constraint coc_nba_pk PRIMARY KEY (n_bastidor)
) engine=innodb;
CREATE TABLE VENTA (
  cod_emp INTEGER NOT NULL,
  dni VARCHAR(9),
  n_bastidor VARCHAR(17),
  any DATE NOT NULL,
  preu INTEGER NOT NULL,
  constraint ven_nba_pk PRIMARY KEY  (n_bastidor,dni),
  constraint ven_cod_fk FOREIGN KEY (cod_emp) REFERENCES EMPLEAT (cod_emp),
  constraint ven_dni_fk FOREIGN KEY (dni) REFERENCES CLIENTE (dni),
  constraint ven_nba_fk FOREIGN KEY (n_bastidor) REFERENCES COCHE (n_bastidor),
  constraint ven_pre_ck check (preu>0)
) engine=innodb;
```

5. EXERCICI 5

Afig la columna **jutge** de tipus cadena(50) a la taula **Plet** de l'exercici 3.

```
ALTER TABLE PLET ADD (jutge) VARCHAR(50);
-- funciona també sense el parèntesi després d'ADD
```

6. EXERCICI 6

Afig la columna **nom** de tipus cadena(50) a la taula **Empleat** de l'exercici 4.

```
ALTER TABLE EMPLEAT ADD nom VARCHAR(50);
```

7. EXERCICI 7

Modifica la columna **preu** de la taula **Venda** de l'exercici 4 perquè el seu tipus ara siga coma flotant. Mantingues les restriccions que tinguera.

```
ALTER TABLE PLET ADD (jutge) VARCHAR(50);
-- funciona també sense el parèntesi després d'ADD
```

8. EXERCICI 8

Esborra la columna **director** de la taula **Pel·lícules** de l'exercici 1.

```
ALTER TABLE PELICULES DROP director;
```

9. EXERCICI 9

Esborra la taula **Pel·lícules** de l'exercici 1.

```
DROP TABLE PELICULES;
```

10. EXERCICI 10

Afig la restricció **habitants > 0** a la taula **Ciutat** creada en l'exercici 2.

```
ALTER TABLE CIUTAT ADD CONSTRAINT ciu_hat_CK CHECK (habitants > 0);
```

També:

```
ALTER TABLE CIUTAT ADD CHECK (habitants > 0);
-- sense nom a la restricció
```